

Theoretische Informatik und Logik

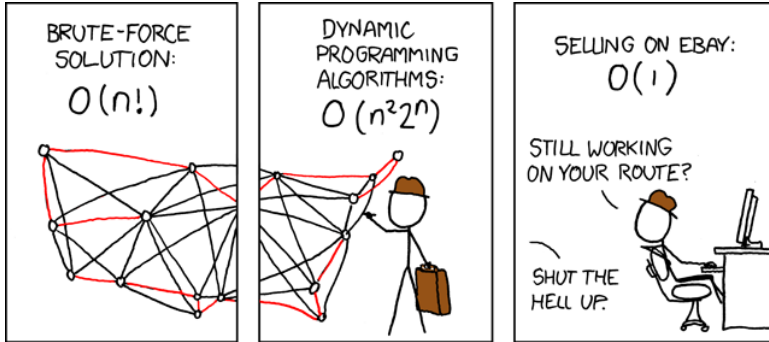
25. Vorlesung: Zusammenfassung und Ausblick

Markus Krötzsch

Professur Wissensbasierte Systeme

TU Dresden, 23. Juli 2026

Zusammenfassung



Randall Munroe, <http://xkcd.com/399/>, CC-BY-NC 2.5

Kernthemen der Komplexität

Methoden

- Komplexitätsklassen
- Many-One Reduktionen (polyzeit, logspace)
- Schwere & Vollständigkeit
- Kodierung und Pseudopolynomialität

Konkrete Klassen und vollständige Probleme

- L
- NL: s-t-Erreichbarkeit (in gerichteten Graphen), ...
- P: **HornSAT**, ...
- NP: **SAT**, **Clique**, **SubsetSum**, ...
- PSPACE: **TrueQBF**, **Geography**, prädikatenlogisches Model Checking, ...
- EXPTIME: Schließen in Datalog, verallgemeinertes Schach, ...

bei NP und PSPACE jeweils mit konkreten Reduktionen zum Beweis der Vollständigkeit

Zwei Hierarchien

Wir haben zwei Hierarchien von Sprachklassen kennengelernt

(1) Chomsky-Hierarchie:

$$\text{Typ 3} \subset \text{det. kontextfrei} \subset \text{Typ 2} \subset \text{Typ 1} \subset \text{Typ 0}$$

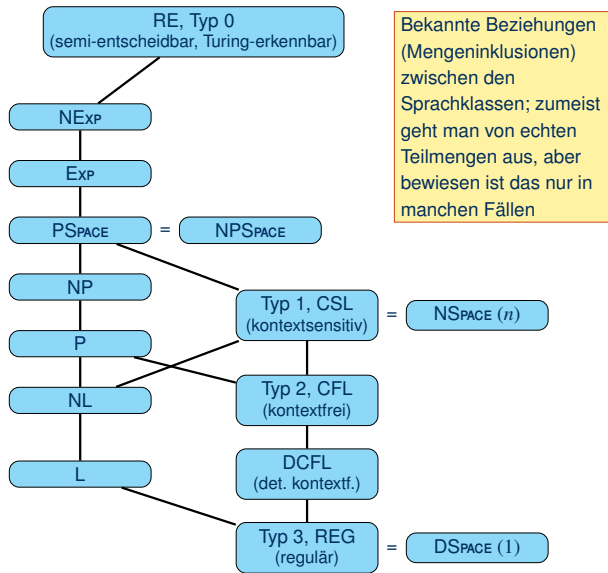
Ansatz: natürliche Definition über Grammatiken

(2) Hierarchie der Komplexitätsklassen:

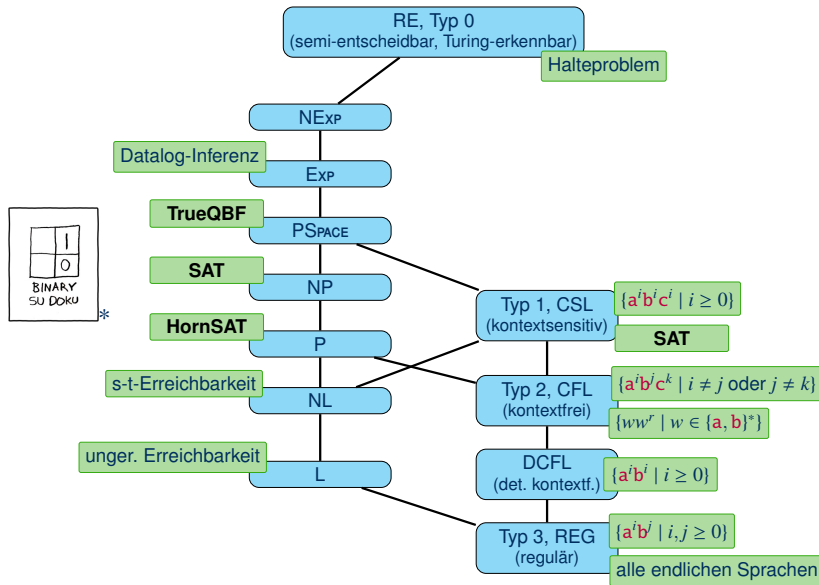
$$L \subseteq NL \subseteq P \subseteq NP \subseteq PSPACE \subseteq EXP \subseteq NEXP$$

Ansatz: natürliche und robuste Definition durch Beschränkung von Turingmaschinen

Eine Hierarchie?

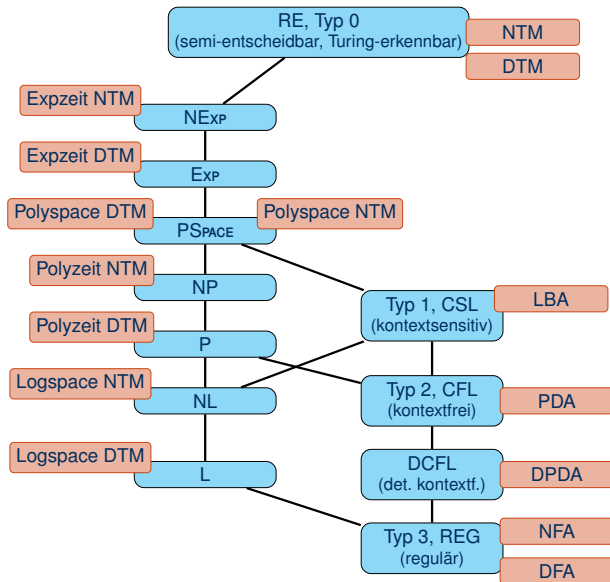


Typische Beispiel-Sprachen



* Randall Munroe, <http://xkcd.com/74/>, CC-BY-NC 2.5

Berechnungsmodelle

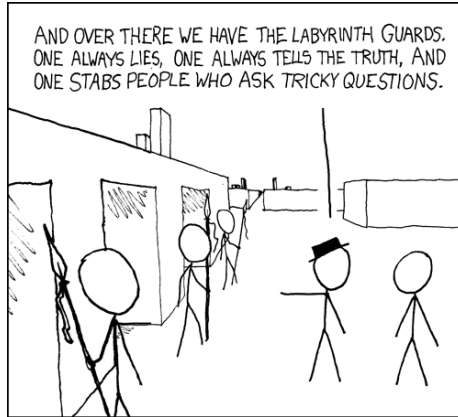


Ein einfacher Beweis für $P = NP$;-)

Wir wissen:	$L \in P$	impliziert	$L \in NP$
Daher gilt:	$L \notin NP$	impliziert	$L \notin P$
Anders gesagt:	$L \in coNP$	impliziert	$L \in coP$
Das heißt:	$coNP \subseteq coP$		
Wegen $coP = P$ gilt:	$coNP \subseteq P$		
Somit gilt:	$NP \subseteq P$		
D.h., wegen $P \subseteq NP$ gilt:	$NP = P$		

q.e.d.?

Logik



Randall Munroe, <http://xkcd.com/246/>, CC-BY-NC 2.5

Kernthemen der Logik

Aussagenlogik

- Syntax und Semantik
- Äquivalenzen und Normalformen
- Logisches Schließen: Resolution und DPLL

QBF

Prädikatenlogik

- Syntax und Semantik
- Äquivalenzen und Normalformen
- Skolemisierung und Herbrandmodelle
- Logisches Schließen: Resolution (einschließlich Unifikation)
- Unentscheidbarkeit des logischen Schließens

+ Vollständigkeit, Korrektheit und Terminierung von Algorithmen

Spezialthemen der Logik

Datenbanktheorie

- Datenbanken als endliche Interpretationen
- Anfragen als Formeln (mit freien Variablen)
- Anfragebeantwortung als Model Checking

Logikprogrammierung

- Horn-Logik (in Aussagen- und Prädikatenlogik)
- Datalog
- Anfragebeantwortung als prädikatenlogisches Schließen oder Second-Order Model Checking

+ relevante Entscheidbarkeiten & Komplexitäten

Informatik ist überall dort, wo gerechnet wird

Rechnen = Schlussfolgern

Beweisen – Nachvollziehen – Verstehen

Ausblick

Idee (aus Sicht der Prädikatenlogik):

- Verwende genau ein zweistelliges Prädikat r und ansonsten nur einstellige Prädikate
- Beschränke Quantifikation in Formeln $\mathbf{F}$ syntaktisch:

$$\mathbf{F}[x] ::= p(x) \mid (\mathbf{F}[x] \wedge \mathbf{F}[x]) \mid (\mathbf{F}[x] \vee \mathbf{F}[x]) \mid \neg \mathbf{F}[x] \mid \exists y. r(x, y) \wedge \mathbf{F}[y] \mid \forall y. r(x, y) \rightarrow \mathbf{F}[y]$$

- ~ immer genau eine freie Variable pro Formel
- ~ immer nur zwei Variablen nötig (Übung: warum?)
- ~ variablenfreie Kurzschreibweise:

$$\mathbf{F} ::= p \mid (\mathbf{F} \wedge \mathbf{F}) \mid (\mathbf{F} \vee \mathbf{F}) \mid \neg \mathbf{F} \mid \diamond \mathbf{F} \mid \square \mathbf{F}$$

Modallogik: Was bringt das?

Intuition:

- Interpretationen sind Graphen mit einer **Erreichbarkeitsrelation** r (sogenannte *Kripke-Strukturen*)
- Knoten des Graphen sind “**mögliche Welten**” in denen jeweils unterschiedliche aussagenlogische Propositionen gelten können (“ $p(x)$ ” bedeutet “in Welt x gilt p ”)
- Formeln werden von einer Welt aus ausgewertet: “Was ist von hier aus möglich?”

Beispiel	Bedeutung von r	Bedeutung von $\Diamond p$	Bedeutung von $\Box p$
Alethische Logik	könnte sein	p ist vorstellbar	p ist notwendig
Temporallogik	passiert später	p gilt irgendwann	p gilt immer
Epistemische Logik	aus Sicht (eines Agenten) gleich	Agent hält p für möglich	Agent weiß p
Dynamische Logik	Erreichbarer Zustand (eines Programms)	p nach Programmende möglich	p nach Programmende sicher

Und viele weitere mehr ...

Modallogik: Eigenschaften und Erweiterungen

Hauptvorteil von Modallogiken: Einfacher als Prädikatenlogik, logisches Schließen in der Regel entscheidbar.

Viele Erweiterungen und Spezialisierungen:

- Einschränkung der Form der betrachteten Erreichbarkeitsgraphen (Interpretationen), z.B. transitive Erreichbarkeit in Temporallogiken oder reflexive Erreichbarkeit in epistemischen Logiken
- Betrachtung von mehreren Erreichbarkeitsrelationen (multimodale Logiken, insbesondere viele Beschreibungslogiken)
- Zusätzliche Modalitäten, z.B. in Temporallogik “ p gilt als nächstes” oder “ p gilt solange bis irgendwann q gilt”
- ...

Relevante Lehrveranstaltungen: **Modal Logic**, **Description Logics**, ...

Logik höherer Ordnung

Prädikatenlogik ist genau genommen **Prädikatenlogik erster Stufe**.

Hintergrund:

- Erste Stufe: Quantoren beziehen sich auf Domänenelemente

Beispiel: „Jede natürliche Zahl n hat einen Nachfolger $s(n)$ “

- Zweite Stufe: Quantoren beziehen sich auf Prädikate

Beispiel: „Für jede Menge M gilt: Enthält M die Zahl 0 und mit jeder natürlichen Zahl n auch stets deren Nachfolger $s(n)$, so enthält M alle natürlichen Zahlen.“

Logik zweiter Ordnung:

- Ausdrucksstärker; kann z.B. die natürlichen Zahlen exakt charakterisieren
- Schwieriger: kein vollständiges und korrektes Beweisverfahren

↪ siehe Vorlesung **Advanced Logic**

Komplexitätstheorie

Wesentliche Grundlagen der klassischen Komplexitätslehre wurden hier schon behandelt

Weiterführende Themen (Beispiele):

- Hierarchietheoreme: Kann man mit mehr Zeit/Speicher wirklich mehr berechnen?
- Relative Komplexität: Orakel
- Komplexitäten unterhalb von P: Schaltkreise als Rechenmodell
- Rechnen mit Zufall: Randomisierte Komplexität
- Einführung in Quantencomputer
- ...

↪ siehe Vorlesung **Complexity Theory** (Wintersemester)

Die Theorie der Datenbanken ist ein wichtiges Anwendungsgebiet für viele Themen aus theoretischer Informatik und Logik

Weiterführende Themen (Beispiele):

- Anfragesprachen vergleichen bzgl. Komplexität und Ausdrucksstärke
- Relationales Kalkül (=Prädikatenlogik)
- Datalog: rekursive Anfragesprache, Fragment der Logik zweiter Stufe
- Graph-Anfragen: Erreichbarkeit und Co. berechnen
- Anfragen unter Berücksichtigung von Constraints
- ...

↪ siehe Vorlesung **Database Theory** (Sommersemester)

Programm- und Hardwareverifikation ist eine wichtige Anwendung logischer Methoden

Weiterführende Themen (Beispiele):

- Modellierung reaktiver Systeme: Transaktionssysteme
- Automatenmodelle zur Darstellung verifizierbarer Eigenschaften
- Lineare Temporallogik (LTL) und Computation Tree Logic (CTL)
- Probabilistische und zeitgesteuerte Automaten
- ...

↪ siehe Vorlesung **Model Checking**

Symbolische Wissensrepräsentation

Anwendungsgebiet der formalen Logik, bei dem menschliches Wissen logisch kodiert und automatisch ausgewertet wird

Weiterführende Themen (Beispiele):

- Entwicklung logischer Sprachen, für die Schlussfolgerung (effizient) entscheidbar ist
- Meist durch Einschränkung auf Teilmengen der Prädikatenlogik, z.B. bei Beschreibungslogiken
- Ontologien: logische Wissensmodelle
- Entwicklung effizienter Ableitungsalgorithmen und Implementierungen
- ...

↪ verschiedene Vorlesungen, z.B. **Description Logics**

Semantic Web & Datenaustausch

Anwendungsgebiet zwischen Datenbanken, Wissensrepräsentation und Webtechnologie

Weiterführende Themen (Beispiele):

- Standards zur Kodierung von Fakten und Schemainformationen: RDF, OWL, ...
- Informationsintegration in (Web)Graphdatenbanken
- Anfragesprachen: SPARQL und ontologiebasierte Anfragen
- Anwendungen (z.B. Wikidata)
- ...

→ siehe Vorlesungen **Knowledge Graphs** (Wintersemester), **Semantic Web Technologies**

Forschung (und studentische Arbeiten)

Wissensrepräsentation: Wie kann menschliches Wissen digital dargestellt werden?

- Herausforderung: Flexibilität vs. Nutzbarkeit
- Herausforderung: Ausdrucksstärke vs. Verarbeitungskomplexität
- Schwerpunkt: logische Ontologie- und Anfragesprachen (Regeln, DLs, Graph-Queries, ...)

Wissensverarbeitung: Wie kann dieses Wissen automatisch genutzt werden?

- Herausforderung: theoretische Komplexität $\neq$ praktische Performance
- Herausforderung: prinzipielles Verfahren $\neq$ implementierbarer Algorithmus
- Schwerpunkt: Logisches Schließen, Anfragebeantwortung über großen Daten

Wissensmanagement: Wie können diese Ergebnisse praktisch genutzt werden?

- Herausforderung: konkrete Märkte und Nutzer?
- Herausforderung: benötigt mehr als eine Kerntechnologie
- Schwerpunkt: Wissensgraphen, Wikidata, Ontologien für LLMs und Co.

Beispiele für BSc/MSc-Themen

Wissensrepräsentation: Wie kann menschliches Wissen digital dargestellt werden?

- Weiterentwicklung von Methoden der logischen Modellierung
- Analyse von Komplexität und Ausdrucksstärke
- Verallgemeinerung neuer Ideen unter Anleitung

Wissensverarbeitung: Wie kann dieses Wissen automatisch genutzt werden?

- Regelengine Nemo: neue Ideen implementieren und testen
- Logische Modelle analysieren und automatisch verstehen
- Ansätze verknüpfen und Ideen übertragen

Wissensmanagement: Wie können diese Ergebnisse praktisch genutzt werden?

- Erklärbarkeit und Verständlichkeit: UIs, Bedienkonzepte, ...
- Große Wissensgraphen nutzen: Qualität, Aufteilung, ...
- Neue Anwendungen: Wissensgraphen in LLMs, räumlich-zeitliche Modelle, ...

Fragen?

Was erwartet uns als nächstes?

- Angestrenzte Klausurvorbereitung (inklusive Probeklausur)
- Betreute Lernräume
- Klausur